

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Andrej Žefran

**Mobilna aplikacija za inventurni popis**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2014



UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Andrej Žefran

**Mobilna aplikacija za inventurni popis**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Rok Rupnik

Ljubljana, 2014



To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuirajo predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani [creativecommons.si](http://creativecommons.si) ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco *GNU General Public License*, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuirajo in/ali predelujejo pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses>.



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Izvajanje inventure z uporabo papirja je izjemno naporno in časovno potratno opravilo. V ta namen so se na trgu pojavile posebne robustne mobilne naprave, ki omogočajo namestitev mobilne aplikacije in imajo čitalec črtne kode.

Zasnujte in razvijte mobilno aplikacijo za inventurni popis s črtno kodo. Aplikacijo razvijte z orodjem Microsoft Visual Studio za platformo Windows Mobile.

MENTOR: doc. dr. Rok Rupnik





## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Andrej Žefran, z vpisno številko **63980189**, sem avtor diplomskega dela z naslovom:

*Mobilna aplikacija za inventurni popis*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Roka Rupnika,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 17. september 2014

Podpis avtorja:



*Za mentorstvo in pomoč pri izdelavi diplomske naloge se zahvaljujem mentorju doc. dr. Roku Rupniku. Zahvalil bi se tudi svojim staršem, ki so mi omogočili študij in me pri tem podpirali. Prav tako bi se zahvalil vsem sošolcem in prijateljem, ki so mi v času študija na kakršenkoli način pomagali, me spodbujali in tudi priganjali.*

*Moji dragi Marjetki.*

# Kazalo

**Povzetek**

**Abstract**

**Poglavje 1 Uvod .....19**

**Poglavje 2 Tehnologije, razvojna in programska orodja .....3**

2.1 Črtna koda .....3

2.1.1 Namen uporabe .....3

2.1.2 Branje črtne kode .....3

2.1.3 Tipi črtnih kod .....4

2.2 Strojna in programska oprema .....5

2.2.1 Ročni računalniki oz. mobilni terminali .....5

2.2.2 Mobilni operacijski sistemi .....6

2.2.3 DataWedge .....7

2.3 Razvojno okolje .....7

2.3.1 Microsoft Visual Studio .....7

2.3.2 Programski jezik C# .....8

2.3.3 .NET in njegovo ogrodje .....8

2.3.4 Java EE (IDE) in Eclipse .....11

2.3.5 MySQL .....12

2.3.6 Web service .....13

2.4 Kriptografija .....13

2.4.1 Kriptiranje in principi moderne kriptografije .....13

**Poglavje 3 Inventura oz. popis artiklov .....15**

**Poglavje 4 Razvoj mobilne aplikacije .....19**

4.1 Nekaj sklepov za razvoj aplikacije .....19

---

4.2	Zahteve aplikacije.....	19
4.3	Arhitektura.....	21
4.4	Struktura podatkovne baze .....	22
4.4.1	Tabela »popisnik«.....	22
4.4.2	Tabela »lokacije« .....	23
4.4.3	Tabela »popis« .....	24
4.5	Kriptiranje gesla .....	25
4.6	Spletni vmesnik .....	25
4.7	Predpogoji za delovanje aplikacije.....	25
<b>Poglavje 5</b>	<b>Uporaba mobilne aplikacije .....</b>	<b>27</b>
5.1	Testiranje aplikacije .....	27
5.2	Nastavitev povezave do baze podatkov.....	28
5.3	Prijava v aplikacijo .....	29
5.4	Glavni zaslon »Popis«.....	30
5.4.1	Popis artikla na izbrano lokacijo .....	31
5.4.2	Ponoven popis istega artikla .....	32
5.5	Pregled seznama popisa.....	33
5.6	Šifrant lokacij .....	34
5.6.1	Dodajanje in urejanje lokacije.....	35
5.7	Šifrant uporabnikov .....	36
5.7.1	Dodajanje in urejanje uporabnika .....	37
<b>Poglavje 6</b>	<b>Sklepne ugotovitve .....</b>	<b>39</b>

## Kazalo slik

Slika 2.1 Primer črtne kode tipa CODE 128 .....	4
Slika 2.2 Ročni terminal Symbol MC9090K (levo) in Motorola MC9590 (desno).....	5
Slika 2.3 Pregled platforme .NET s komponentami ogrodja .NET .....	9
Slika 2.4 Izvleček iz pregleda platforme .NET .....	10
Slika 2.5 Kriptografske komponente .....	14
Slika 2.6 Zgoščevalne funkcije (ang. hash) .....	14
Slika 4.1 Primer zahtevane črtne kode .....	20
Slika 4.2 Arhitektura aplikacije .....	21
Slika 4.3 Struktura tabele popisnik .....	22
Slika 4.4 Struktura tabele lokacije .....	23
Slika 4.5 Struktura tabele popis .....	24
Slika 5.1 Prijavni zaslon (levo) in zaslon nastavitev (desno) .....	28
Slika 5.2 Prijava v aplikacijo .....	29
Slika 5.3 Prikaz zaslona oz. menija uporabnika (levo) in skrbnika (desno).....	30
Slika 5.4 Izbor lokacije (levo) in primer popisa (desno) .....	31
Slika 5.5 Obvestilo o ponovnem popis že popisane artikla .....	32
Slika 5.6 Seznam popisa .....	33
Slika 5.7 Šifrant lokacij .....	34
Slika 5.8 Izbrana lokacija .....	35
Slika 5.9 Šifrant uporabnikov .....	36
Slika 5.10 Izbrani uporabnik .....	37

## Seznam uporabljenih kratic

kratica	angleško	slovensko
<b>ASCII</b>	American standard code for information interchange	Ameriški standardni nabor za izmenjavo informacij
<b>Code128</b>	barcode standard	standard črtne kode
<b>EE</b>	enterprise edition	nekakšna večja izdaja
<b>GPS</b>	global positioning system	
<b>HTTP</b>	hypertext transfer protocol	protokol za prenos informacij na spletu
<b>ID</b>	identity	identiteta
<b>IDE</b>	integrated development environment	integrirano razvojno okolje
<b>IP</b>	internet protocol	internetni protokol
<b>IP Code</b>	international protection marking (ingress protection)	mednarodna standardna oznaka zaščit (zaščita pred vdorom v opremo)
<b>IR Blaster</b>	infrared blaster	naprava, ki simulira infrardečo daljinsko povezavo
<b>JDK</b>	java development kit	javanski razvojni paket
<b>MySQL</b>	my structured query language	sistem za upravljanje s podatkovnimi bazami
<b>NFC</b>	near field communication	visokofrekvenčna komunikacijska tehnologija kratkega dosega
<b>OS</b>	operating system	operacijski sistem
<b>SDK</b>	structured development kit	komplet orodij za razvijanje programske opreme
<b>SQL</b>	structured query language	strukturni povpraševalni jezik
<b>USB</b>	universal serial bus	univerzalno serijsko vodilo
<b>WLAN (Wi-Fi)</b>	wireless local area network	brezžično lokalno omrežje



## Povzetek

V diplomski nalogi je predstavljena mobilna aplikacija za inventurni popis. Aplikacija je namenjena ročnim računalnikom. Omogoča učinkovit inventurni popis artiklov označenih s črtno kodo. Po mnogih podjetjih še vedno izvajajo ročen inventurni popis, katerega je z današnjo tehnologijo, že dlje časa možno delno, ali v celoti zamenjati z elektronskim popisom. Elektronski način je za razliko od ročnega enostavnejši, zanesljivejši ter hitrejši.

Mobilna aplikacija je razvita s programskim orodjem Microsoft Visual Studio, za operacijski sistem Windows Mobile. Omogoča shranjevanje popisnih podatkov, z možnostjo večjega števila popisovalcev hkrati, na istih ali pa različnih lokacijah. Podatke aplikacija shranjuje v skupno podatkovno bazo MySQL na oddaljenem lokalnem ali pa spletnem strežniku. V nalogi so predstavljene tudi zahteve za izdelavo aplikacije, razvoj le te in prikaz načina uporabe.

**Ključne besede:** inventura, popis, mobilna aplikacija, mobilni računalnik, ročni terminal, črtna koda, Windows Mobile, Microsoft Visual Studio, MySQL



## Abstract

The thesis focuses on a mobile application for inventory management that runs on a mobile device. The app is intended for handheld computers and enables an effective inventory of the products. Many companies still conduct physical inventories, even though this method has long become redundant thanks to today's technology and can be partially or completely replaced with electronic inventory systems. Unlike physical inventories, electronic inventories are simple, reliable, and quicker.

The mobile application has been developed using the Microsoft Visual Studio software tool and is intended for the operating system Windows Mobile. It enables the storage of information acquired through inventories conducted by multiple employees at the same time and on the same or different locations. The app stores information in MySQL database, on LAN or Web server. The thesis also focuses on requirements for the making of the application and on its development, as well as it demonstrates its usage.

**Keywords:** inventory, mobile application, mobile device, handheld computer, Pocket PC, barcode, Windows Mobile, Microsoft Visual Studio, MySQL

## Poglavje 1      Uvod

Z inventurnim popisom se danes srečujemo že v vseh podjetjih, saj nas tako zakon o gospodarskih družbah kot tudi želja po dobrem poslovanju silita v redno izvajanje le teh. V inventurni popis vključujemo od zaloge surovin in materiala (primer: v skladiščih), do zalog nedokončane proizvodnje, zalog blaga, denarna sredstva, osnovna sredstva, itd.

V mnogih podjetjih se inventurni popis še vedno izvaja ročno. Tudi na splošno je sam postopek prepoznavanja in pregleda artiklov časovno zelo potraten, pri ročnem popisu pa tudi nepregleden in nezanesljiv ter se za ta namen porabi mnogo preveč dragocenega časa. Večkrat prihaja tudi do slabe volje zaposlenih, saj se pogosto zgodi, da se številke med knjižnim stanjem v poslovnih knjigah in dejanskim stanjem ne ujemajo.

Da bi se ročnega popisa znebili in popisovalcem in zaposlenim olajšali delo ter tako privarčevali na izgubljenem času smo se odločili, da pripravimo rešitev z razvojem mobilne aplikacije, ki bi omogočila hiter, učinkovit in zanesljiv inventurni popis.

Ker pa se tehnologija razvija in danes obstaja na trgu kar nekaj različnih mobilnih naprav, z različnimi operacijskimi sistemi smo se v nalogi osredotočili na določene pogoje in izbrali ustrezno strojno opremo. Tako smo se odločili za uporabo robustnih ročnih terminalov, ki omogočajo delo v vseh delovnih pogojih ter imajo vgrajen bralnik črtne kode. Črtna koda tudi zato, ker je že nekaj časa zelo razširjena in bo posledično tudi mobilna aplikacija uporabna na širšem področju.

V poglavju Tehnologije, razvojna in programska orodja bomo spoznali opremo in tehnologije z katerimi smo se spoznali pri razvoju, testiranju in uporabi mobilne aplikacije. Poglavje 3 na kratko opisuje podrobnosti o inventurnem popisu. Poglavje 4 bo predstavilo razvoj in arhitekturo mobilne aplikacije. V Poglavje 5 se bomo srečali še s predstavitvijo in uporabo mobilne aplikacije. Sledil pa mu bo samo še sklepni zaključek.

## **Poglavje 2     Tehnologije, razvojna in programska orodja**

### **2.1   Črtna koda**

Črtna koda je neke vrste elektronski ključ. Sestavljena je iz niza temnih črt in svetlih presledkov. Slika 2.1 prikazuje enega od tipov črtne kode, ki jo srečamo v praksi.

#### **2.1.1 Namen uporabe**

Najpogosteje se uporablja v maloprodaji na embalaži, in sicer za zapis številke artikla. Ta omogoča enostaven, hiter in zanesljiv strojni prenos podatka s tiskanih dokumentov, nalepk, embalaže v razno računalniško opremo in blagajne. S časoma so se razvili različni standardi oz. tipi črtnih kod, ki so omogočili, da se je uporaba le teh razširila tudi na druga področja. [10]

#### **2.1.2 Branje črtne kode**

Črtna koda je čitljiva z bralniki črtne kode. Bralniki so lahko samostojne naprave (kot ročni računalniki oz. terminali), ki imajo vgrajen pomnilnik in program. Takšni bralniki omogočajo hiter popis inventure. Največkrat pa se srečamo z bralniki s samo nameščenim programom, ki dešifrira črtno kodo (spremeni jo v niz števil in črk). Ta program ugotovi kateri tip kode je bil uporabljen in ga nato pretvori. Takšne bralnice priključimo, recimo preko USB vmesnika na osebni računalnik. Če bralnik ne zna prebrati črtne kode, zaradi poškodbe le te ali česar koli drugega, se lahko podatek vnese z uporabo tipkovnice. Ravno zaradi tega razloga je ponavadi nekje ob črtni kodi vedno izpisan še niz znakov, ki je zakodiran vanjo. [5]

### 2.1.3 Tipi črtnih kod

V osnovi delimo na linearne (1D) in sestavljene (2D) tipe črtnih kod. Med seboj se ti dve razlikujejo predvsem v treh ključnih elementih:

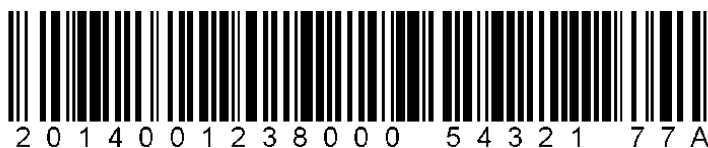
- **količina podatkov** - neprimerno večja v 2D kodah
- **velikost oz. površina kode** - v primeru 2D kod je ta veliko manjša
- **način zajema podatkov** - ob upoštevanju uporabe zmogljivejših bralnikov oz. terminalov z integrirano kamero je pri 2D kodah veliko hitrejši. [4]

#### 2.1.3.1 Uporabljen črtna koda tipa »CODE 128«

Za pripravo in natis črtnih kod oz. za testiranje aplikacije smo uporabili črtno kodo tipa CODE 128 (Slika 2.1). Ta omogoča alfa-numerični zapis 128 znakov v načinu ASCII, izjemno zgoščenost zapisa in tri različne nabore znakov:

- A (kombinacija števil in samo malih črk)
- B (kombinacija števil, malih in velikih črk)
- C (samo številke, bolj zgoščen zapis).

Te črtne kode najpogosteje srečujemo pri označevanju osnovnih sredstev. Za zajem podatkov v tovrstnih kodah se priporočajo ročni računalniki z vgrajenimi bralniki črtnih kod. [18]



Slika 2.1 Primer črtne kode tipa CODE 128

## 2.2 Strojna in programska oprema

### 2.2.1 Ročni računalniki oz. mobilni terminali

Ročni računalniki, pogosto imenovani tudi kot mobilni ali ročni terminali so pri hitrem premagovanju poslovnih odločitev nepogrešljivi. So zmogljivi in lahki ter robustni, mnogi delujejo tudi pri ekstremnih temperaturah od -20°C do + 50°C. Odporni so na vlago in prah ter na udarce (IP 54, IP 64, IP 65 in IP 67). Hkrati so enostavni za uporabo in se dobro odrežejo v skladiščih, trgovini, proizvodnji, hladilnicah, terenski prodaji, na letališčih itd. Vsebujejo zmogljive procesorje in velike pomnilnike. Njihovi grafični prikazovalniki so praviloma občutljivi na dotik. Vsebujejo tudi tipkovnico, kontrolni indikator, vibrator ter seveda laserske čitalnike. [16]

Ročne računalnike lahko razdelimo tudi glede na operacijski sistem:

- ročni računalniki z MS Windows operacijskim sistemom
- drugi ročni računalniki

#### 2.2.1.1 Ročni terminal Motorola serija MC9000

Eden izmed najbolj razširjenih in zelo kvalitetnih ročnih terminalov je vsekakor terminal znamke Symbol ali po novem Motorola serija MC9000. Imeli smo priložnost testiranja aplikacije tako na starejšemu, zelo razširjenemu modelu MC9090K, kot tudi na najnovejši različici te serije MC9590. Oba terminala sta prikazana na Slika 2.2.



Slika 2.2 Ročni terminal Symbol MC9090K (levo) in Motorola MC9590 (desno)

Ročne računalnike *Symbol* oz. *Motorola* odlikujejo prijazna oblika, velik zaslon in robustnost ter dodatna zaščita ohišja, ki omogoča delo v vseh delovnih pogojih. Ti terminali sodijo med najodpornejše terminale vseh časov, saj so odporni na prah in vlago, padce, udarce, ekstremne temperature itd. Izbiramo lahko med več tipi skenerjev, ki so prilagojeni različnim potrebam zajema podatkov. Črtne kode zaznavajo z razdalje od 10 cm do 12 m, kar pomeni, da lahko odčita črtne kode tudi iz viličarja ali v skladišču z zelo visokimi regali. Velik zaslon, občutljiv na dotik, ki omogoča vnos podatkov tudi preko prikazovalnika, pomeni enostavno delo in hiter vnos podatkov tudi v zatemnjenih prostorih. Izbirati je mogoče tudi med tipkovnicami z različnim številom tipk. [17]

Novejši model MC9590 je še bolj izpopolnjen z boljšimi specifikacijami, kot so močnejši procesor, več pomnilnika, več prostora za shranjevanje, pametno baterijo, vizualno lepši, z novejšim operacijskim sistemom *Windows Mobile 6.5* ter z večjo ločljivostjo zaslona 640x480. [14]

## 2.2.2 Mobilni operacijski sistemi

Mobilni operacijski sistem je sistem, ki nadzoruje in upravlja mobilno napravo. Med mobilne naprave štejemo: pametne telefone, tablične računalnike, dlančnike in razne druge. Mobilni operacijski sistemi so proti klasičnim enostavnejši in uporabljajo brezžično komunikacijo. Sodobni mobilni operacijski sistemi združujejo funkcije operacijskega sistema osebnega računalnika z drugimi funkcijami, vključno z zaslonom na dotik, Bluetooth, Wi-Fi, GPS mobilno navigacijo, fotoaparatom, video kamero, prepoznavanjem govora, snemalnikom, predvajalnikom glasbe, NFC in infrardečo povezavo. Ker smo našo aplikacijo razvili za operacijski sistem *Microsoft Windows Mobile*, bomo ta sistem v nadaljevanju tudi na kratko opisali. [13]

### 2.2.2.1 Microsoft Windows Mobile

*Windows Mobile* je družina mobilnih operacijskih sistemov razvita z strani podjetja *Microsoft*, za pametne telefone in naprave *Pocket PC*. *Windows Mobile* je podoben sistemu za osebne in prenosne računalnike z razliko, da je prirejen za mobilne naprave. Njegovi začetki sistema segajo v leto 1996, ko je bil poznan kot *Windows CE* (baziral na kodi *Windows 95*). Nato se je leta 2000 pojavil pod imenom *Pocket PC 2000*, kasneje *Pocket PC 2002*. Prvič pod imenom *Windows Mobile 2003* pa se je pojavil v juniju 2003. Nato so mu sledile verzije *Windows Mobile 2003 SE* (ang. Second Edition, slo. druga izdaja), *Windows Mobile 5*, *Windows Mobile 6*, *Windows Mobile 6.1*, *Windows Mobile 6.5*. Zadnja različica, ki je še izšla po objavi novejšega sistema *Windows Phone* je bila verzija 6.5.5. Po tej različici je podjetje *Microsoft*



razvijalo samo še najnovejši mobilni operacijski sistem *Windows Phone*. Zadnja današnja različica tega sistema je *Windows Phone 8.1*. Operacijski sistem je prešel v širšo rabo z začetki združevanja GSM modula in dlančnika, kar se je kasneje razvilo v danes vsem znani pametni telefon [21].

### 2.2.3 DataWedge

DataWedge je aplikacija oz. pripomoček za bralnike črtnih kod znamke Motorola in Symbol. Ta omogoča napredne zmožnosti za zajem podatkov v vsako aplikacijo, brez da bi za to bilo potrebno programiranje. Teče v ozadju in upravlja vmesnik do obeh, tako vgrajenih, kot tudi priključenih naprav za zajemanje podatkov iz naprave. To so lahko skenerji črtnih kod in druge razne periferne naprave. Zajeti podatki se pretvorijo v znake in se pošljejo v ciljno aplikacijo, kot če bi bili v njo vtipkani z tipkovnico. [3]

Preden smo aplikacijo preizkusili tudi na ročnem terminalu, smo nanj namestili ta pripomoček, in sicer različico 3.3. Program smo nastavili tako, da se zažene ob zagonu operacijskega sistema.

## 2.3 Razvojno okolje

Za razvoj mobilne aplikacije na »*Windows Mobile 5.0*«, smo uporabili razvojno orodje Visual Studio 2008 in komplet orodij »*Windows Mobile 5.0 SDK*« za »*Pocket PC*«. Aplikacijo smo zgradili z programskim jezikom C#, z uporabo *Microsoft .NET Framework 3.5 SPI* programskega ogrodja. Upravljanje baze podatkov smo izvajali z orodjem *phpMyAdmin*. Za razvoj spletnega vmesnika ter komunikacijo z podatkovno bazo pa smo uporabili razvojno orodje *Java EE (Eclipse Luna v 4.4.0)*.

### 2.3.1 Microsoft Visual Studio

*Microsoft Visual Studio* je integrirano razvojno orodje (IDE). Uporabimo ga lahko za razvoj spletnih aplikacij, spletnih strani in spletnih storitev ter za razvoj aplikacij z grafičnim vmesnikom ali brez, tako za namizna, kot tudi mobilna okolja za operacijski sistem *Microsoft Windows* in tudi vse druge platforme, ki poganjajo *.NET Framework*. Omogoča razvoj v mnogih programskih jezikih od *Visual C*, *Visual C++* do *Visual Basic* in drugih.

Glavne značilnosti *Visual Studio*:

- urejevalnik kode z podporo samodejnega zaključevanja programske kode (ang. *Intellisense*), ki tudi vizualno poudari sintakso
- razhroščevalnik (ang. *debugger*) deluje na nivoju izvorne kode in na nivoju strojne kode na kateri poskrbi za lažje testiranje in odkrivanje napak
- za grafični del oblikovanja aplikacije pa imamo na voljo več vrst urejevalnikov v katerih po navadi z metodo »povleci in spusti« oblikujemo in uredimo aplikacijo
- orodjarno
- prevajalnik (orodje za avtomatizacijo)
- pregledovalnik objektov, dokumentov

Razvojno okolje *Visual Studio* nastopa v več različicah, odvisno od namena uporabe: najbolj znane so *Express*, *Standard* in *Professional Edition*. [12]

### **2.3.2 Programski jezik C#**

Programski jezik *C#* je predmetno usmerjen programski jezik, ki ga je razvilo podjetje *Microsoft* in je oblikovan za delo z *.NET Framework* (slo. *.NET platformo*). Je nekakšna kombinacija programskega jezika *C* na *C++* z dodanimi funkcijami iz jezikov *Visual Basic* in *Java*. Čeprav ta jezik temelji na *C++* vsebuje številne sestavine, ki so značilne za *Java* jezik. Razvijalcem *Visual C#* omogoča razvoj visoko prenosljivih aplikacij. [19]

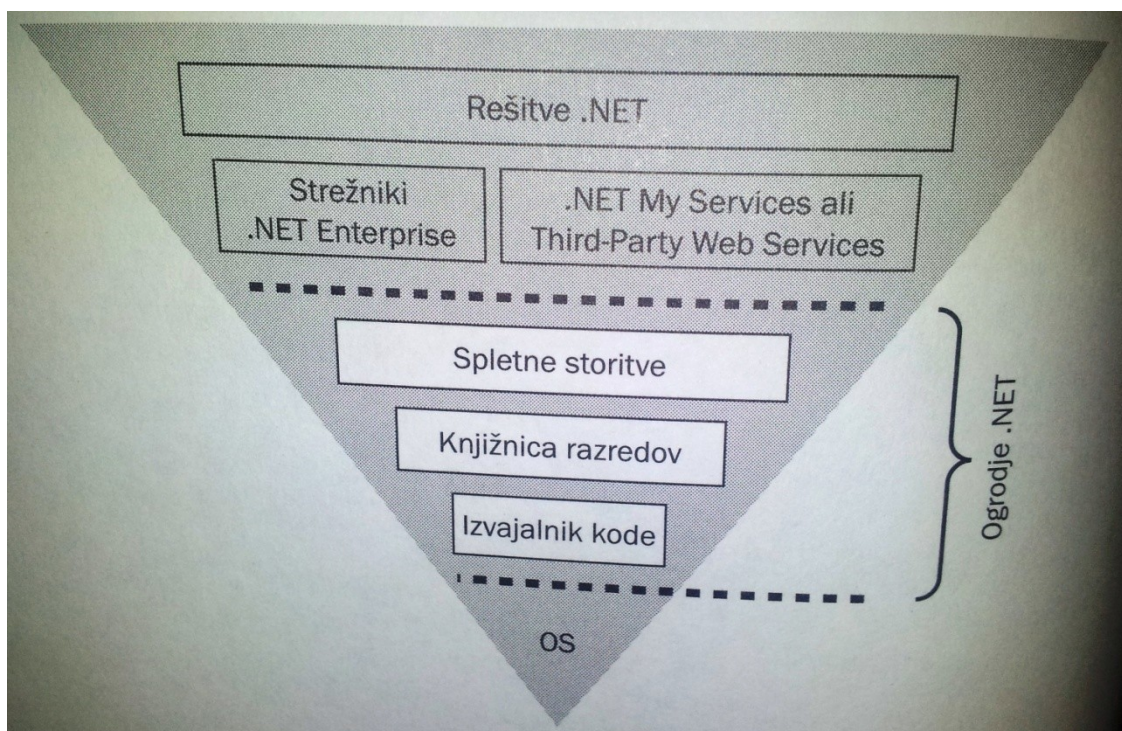
### **2.3.3 .NET in njegovo ogrodje**

*.NET Framework* je večjezično okolje za razvoj, namestitve ter izvajanje *XML* spletnih storitev in aplikacij za vse osebne računalnike, mobilne naprave, itd., ki jo je razvilo podjetje *Microsoft*. Vsebuje je nabor mnogih jezikov (*C#*, *C++*, *VBScript*, *Visual Basic*, *JScript*, *J#* in druge).

Omogoča naslednje funkcionalnosti [22]:

- lažji in hitrejši razvoj
- izvajanje zahtevnih in enostavnih projektov
- možnost kreiranja projektov, sestavljenih iz več manjših modulov, zgrajenih iz različnih programskih jezikov
- prenosljivost programske kode

.NET je spojitev ogrodja .NET, storitev, znanih pod imenom .NET My Services, kar vključuje razvojna orodja, kot je npr. Visual Studio, in infrastrukturnih storitev, ki jih zagotavlja družina strežnikov .NET Enterprise. [1]



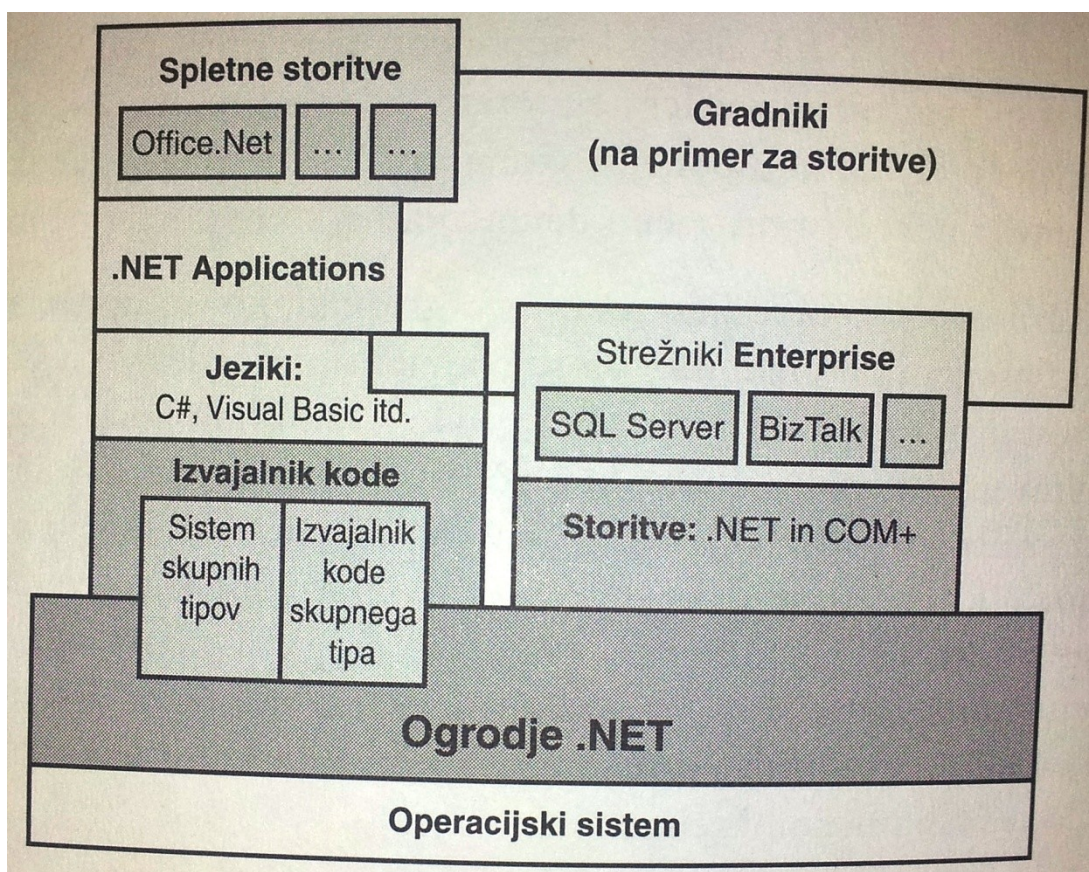
Slika 2.3 Pregled platforme .NET s komponentami ogrodja .NET

### 2.3.3.1 Komponente ogrodja .NET

Ogrodje je razdeljeno na 3 dele (te komponente odločajo kaj je .NET) [1]:

- srce .NET je izvajalnik kode skupnega jezika (ang. CLR – common language runtime)
- knjižnice razredov
- storitve: sistemske (znotraj .NET) in spletne storitve

Razredi in storitve so tisti deli, ki jih razvijalci povežejo v rešitve. Izvajalnik kode skupnega jezika je mehanizem, ki poganja rešitve in omogoča, da lahko uporabljate storitve operacijskega sistema.



Slika 2.4 Izvleček iz pregleda platforme .NET

### **2.3.4 Java EE (IDE) in Eclipse**

Eclipse je skupnost za posameznike in organizacije, ki želijo sodelovati v komercialno prijazni, odprtokodni programski opremi. Njeni projekti so osredotočeni na izgradnjo odprte platforme za razvoj, ki je sestavljena iz delovnih okvirjev, orodja in trajanj izvajanja za izgradnjo, uvajanje in upravljanje programske opreme nad celotnim procesom.

Eclipse Foundation je neprofitna organizacija, ki nudi pomoč uporabnikom in gostuje odprtokodne projekte z imenom »Eclipse Projects«. Kot neodvisna organizacija je bila ustanovljena januarja 2014.

Na splošno velja, da Eclipse Foundation ponuja štiri storitve Eclipse skupnosti:

- infrastrukturo informacijske tehnologije
- IP (Internet Protocol) upravljanje
- razvojne procese
- razvoj ekosistema

Zaposleno osebje je povezano z vsako od teh področij in s pomočjo Eclipse skupnosti pomaga pri zadovoljevanju potreb interesne skupine. [6]

Eclipse IDE je razvojno orodje za Java EE razvijalce spletnih aplikacij, ki vsebuje [9]

- Data Tools Platform
- Eclipse Git Team Provider
- Eclipse Java Development Tools
- Eclipse Java EE Developer Tools
- JavaScript Development Tools
- Maven Integration for Eclipse
- Mylyn Task List



- Eclipse Plug-in Development Environment
- Remote System Explorer
- Eclipse XML Editors and Tools

### 2.3.5 MySQL

*MySQL* je eden najpopularnejših odprtokodnih sistemov za upravljanje s podatkovnimi bazami, ki ga je razvilo podjetje oz. korporacija Oracle. Odprtokodni sistem pomeni, da je vsakemu omogočen dostop do programske opreme in jo lahko tudi prireja za svoje potrebe. Vsak lahko, in sicer brez plačila, z interneta prenese to programsko opremo. Podatkovna baza je strukturirana zbirka podatkov. Lahko je karkoli od preprostega nakupovalnega seznama do zbirke slik ali pa velika količina podatkov v nekem podjetju. Za dodajanje in obdelovanje podatkov v podatkovni bazi potrebujemo *MySQL* strežnik (sistem za upravljanje zbirk podatkov).

*MySQL* je relacijski podatkovni sistem. Relacijska podatkovna zbirka hrani podatke v ločenih tabelah in ne v enem velikem skladišču. Na takšen se močno poveča hitrost in fleksibilnost baze. Logični model, s predmeti, kot so baze podatkov, tabele, vrst pogledov, vrstic in stolpcev, ponuja prilagodljivo programsko okolje.

V imenu *MySQL* se srečamo z kratico *SQL*, ki pomeni strukturiran povpraševalni jezik. *SQL* je najbolj uporabljen standardiziran jezik za dostop do podatkovnih zbirk. Definiran s standardom *ANSI/ISO SQL* se razvija že od leta 1986 in do danes obstaja že mnogo različic.

Programska oprema *MySQL* je zaščiten z licenco *GPL* (*GNU, General Public License*, <http://www.fsf.org/licenses/>) s katero se določa, kaj lahko in česa ne smemo početi s programom v različnih situacijah.

Strežniški sistem *MySQL* je zelo hiter, zanesljiv in prilagodljiv ter enostaven za uporabo. Lahko teče tako na osebni kot tudi prenosni računalnik, poleg vseh vaših ostalih aplikacijah, spletnih strežnikih in zahteva malo oz. skoraj nič pozornosti. Če posvetite celoten računalnik samo za *MySQL* lahko tudi prilagodite nastavitve, da izkoristi celoten pomnilnik in procesorsko moč ter kapaciteto, ki jo ima na voljo. Lahko obsega tudi večje skupine strežnikov, ki so mrežno povezani med seboj. [15]

### 2.3.6 Web service

Web service (slovensko spletna storitev) je metoda, ki omogoča komunikacijo med dvema elektronskima napravama preko omrežja. Je funkcija programske opreme dosegljiva na nekem omrežnem naslovu preko lokalnega ali internetnega omrežja, ki je ves čas na voljo potrebnim napravam. V našem primeru je na voljo uporabnikom, katerim v realnem času omogoča komunikacijo med mobilno napravo in strežnikom z zbirko podatkov.

## 2.4 Kriptografija

Beseda kriptografija ali kriptologija izhaja grščine (kryptos - skrito ali skrivnost, graphein – pisanje oz. besede logia - študij). Je praksa in študija tehnike za varno komunikacijo v prisotnosti tretjih oseb. Na splošno gre za izdelavo in analiziranje protokolov, ki premagujejo vpliv tretjih oseb in se nanašajo na različne vidike informacijske varnosti, kot so zaupnost, integriteta podatkov, preverjanje pristnosti, itd. V tem podpoglavju se bomo dotaknili principov moderne kriptografije, ki danes sega v področja matematike, računalništva in elektrotehnike. Uporabljata se tudi pojma šifriranje (enkripcija) in dešifriranje (dekripcija).

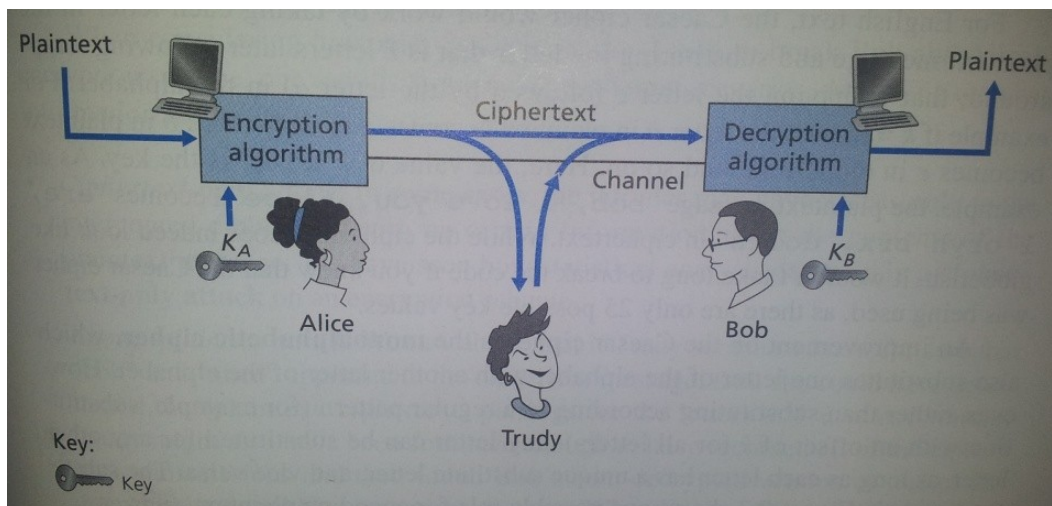
### 2.4.1 Kriptiranje in principi moderne kriptografije

Kriptiranje pomeni skrivanje vsebine. Z kriptiranjem želimo čim bolj otežiti delo potencialnim napadalcem, vdiralcem ali prisluškovalcem. Sporočilo po nekem algoritmu ali metodi šifriramo, ob tem uporabimo določene vrednosti kot parametre v algoritmu, ki jim rečemo ključ. Sogovornika pa se morata dogovoriti o algoritmu in ključu, da lahko komunicirata s šifriranimi sporočili. [11]

Primer uporabe (Slika 2.5): Recimo, da hoče Alice (pošiljatelj) poslati Bob-u (prejemniku) sporočilo. Alice šifrira navadno besedilo (ang. plaintext) z šifrirnim algoritmom, tako da šifrirano sporočilo, poznano tudi kot šifropis (ang. ciphertext), izgleda nerazumljivo vsakemu prisluškovalcu (Trudy) [8].

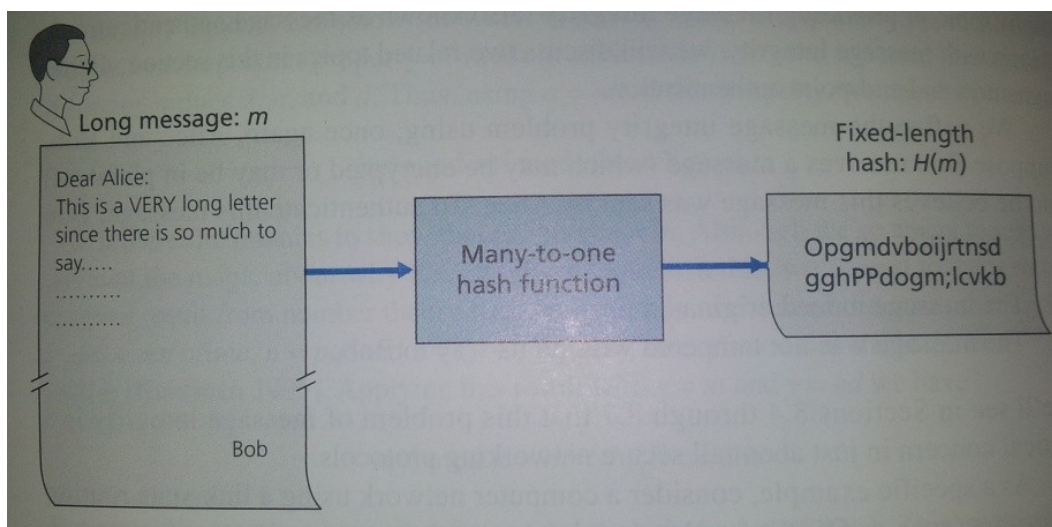
Poznamo dva principa kriptografije:

- kriptografija z simetričnim ključem in
- enkripcija z javnim ključem



Slika 2.5 Kriptografske komponente

Eden izmed mnogih varnostnih mehanizmov, ki smo ga uporabili pri razvoju aplikacije je tudi kriptografska zgoščevalna (ang. hash) funkcija katere splošno delovanje je prikazano na Slika 2.6.



Slika 2.6 Zgoščevalne funkcije (ang. hash)

Zelo pogosto uporabljena kriptografska zgoščevalna funkcija je MD5 (ang. message-digest) hash algoritem. Ta generira 128-bitno vrednost, tipično izraženo v tekstovni obliki besedila kot 32-mestno šestnajstiško število [8].



## **Poglavje 3     Inventura oz. popis artiklov**

Zakon o gospodarskih družbah v 54. členu pravi, da morajo družbe najmanj enkrat na leto preveriti, ali se stanje v poslovnih knjigah ujema z njihovim dejanskim stanjem. Podjetja morajo v skladu s SRS 28.23 (SRS – slovenski računovodski standard), sprejeti notranji pravilnik o popisu, v katerem določijo, kako in kdaj se opravi popis, ter uskladi knjižno stanje z dejanskim.

Pri inventuri oziroma popisu ugotavljamo količinsko stanje sredstev, se pravi blaga na zalogi in virov sredstev, na določen dan in ga primerjamo s stanjem v poslovnih knjigah. Ugotoviti in uskladiti moramo torej razlike med knjižnim stanjem in dejanskim stanjem. Natančni in realni podatki nam posledično prikazujejo tudi kakovostne računovodske informacije, ki kažejo stanje podjetja (višino kapitala, višino poslovnega izida in davčne osnove). Takšni podatki pa lahko vodijo k sprejemanju novih organizacijskih ukrepov in poslovnih politik podjetja, k bolj racionalnemu in odgovornemu poslovanju.

Inventura med zaposlenimi ni najbolj priljubljena, saj se pogosto zgodi, da se številke ne ujemajo, poleg tega pa je sam postopek prepoznavanja in pregleda izdelkov, časovno zelo potraten. Popisuje se sredstva, tudi tista, ki so zunaj podjetja, obveznosti do virov sredstev in sredstva, ki so pri podjetju le začasno (vračljiva embalaža, najem itd.).

Poleg tega lahko v inventuro vključujemo tudi:

- neopredmetena dolgoročna sredstva (naložbe v pridobitev materialnih pravic)
- opredmetena osnovna sredstva (zemljišča, zgradbe, proizvodjalno in drugo opremo, osnovno čredo, večletne nasade in drobní inventar)
- zaloga surovin in materiala (surovine in material v skladišču, surovine in material v dodelavi, predelavi in na poti, drobní inventar in embalaža v skladišču ter drobní inventar in embalaža, dana v uporabo)

- zaloge nedokončane proizvodnje (zalogo nedokončane proizvodnje, zalogo polproizvodov in do konca obračunskega obdobja opravljene storitve, ki jih naročniki še niso prevzeli)
- zaloga blaga
- denarna sredstva na računih
- kratkoročne časovne razmejitve (kratkoročno določene stroške oziroma dohodke)
- popis terjatev in obveznosti
- kratkoročne terjatve
- dolgoročne finančne naložbe
- dolgovi in druge obveznosti
- popis kapitala
- dolgoročne rezervacije

Popis izvaja popisna komisija, ki naj bi vključno s predsednikom imela vsaj tri člane. Med področja dela popisnih komisij pa štejemo:

- ugotavljanje, merjenje, preštevanje, natančnejše opisovanje sredstev ter vpisovanje teh podatkov v popisne liste,
- vpisovanje sprememb, nastalih med dnevom popisa in določenim datumom, pod katerim se popis opravlja,
- vpisovanje dejanskega stanja v popisne liste,
- ugotavljanje razlik med stanjem, ugotovljenim s popisom, in knjigovodskih stanjem,
- dajanje predlogov, kako se te razlike likvidirajo in knjižijo,
- podpisovanje popisnih listin.

Delo komisije je opravljeno, ko v popisu razčisti vse vzroke za razlike med dejanskim in knjigovodskim stanjem. Vsa neskladja se morajo evidentirati v »Poročilu o popisu«, ki je temeljna informacija o obvladovanju in urejenosti podjetja. Vodja popisa organizira popis ter odgovarja za njegovo pravočasnost in pravilnost in na koncu predloži poročilo o popisu pristojnemu organu podjetja. Na podlagi poročila se poslovodstvo podjetja odloči o načinu likvidacije ugotovljenih primanjkljajev, oziroma presežkov materialnih in drugih vrednosti, odpisu in uničenju neuporabnih zastarelih sredstev, odpisu terjatev itd. [7]



## **Poglavje 4     Razvoj mobilne aplikacije**

### **4.1    Nekaj sklepov za razvoj aplikacije**

Pri našem razvoju smo bili osredotočeni bolj na inventuro oz. popis zaloge nedokončane proizvodnje ter na popis zalog surovin in materiala v le tem. Pri takšnih inventurah se največkrat srečujemo z velikim številom artiklov tako v proizvodnjah, kot tudi v raznih skladiščih.

Lokacije v proizvodnjah so ponavadi zelo veliki obsežni prostori. Ti so večkrat ločeni kar po proizvodnih linijah. Na takšnih linijah največkrat ne najdemo oznak lokacij (na primer s črtno kodo) in zato smo pri razvoju aplikacije razmišljali v tej smeri, da za ta namen ne uporabimo bralnika črtne kode. Namesto tega smo se osredotočili, da lokacijo kar izberemo iz spustnega seznama. Lokacije se naložijo v spustni seznam iz šifranta prej vnesenih lokacij oz. nahajališč.

Za nadaljnji razvoj aplikacije pa imamo v mislih tudi branje črtne kode za označena nahajališča z katerimi se srečujemo v večjih skladiščih, kjer najdemo tudi mikrolokacije (lokacije z manjšo površino).

### **4.2    Zahteve aplikacije**

Aplikacijo smo izdelali na podlagi spodnji zahtev:

- Aplikacija naj omogoča hiter, zanesljiv in učinkovit inventurni popis zaloge nedokončane proizvodnje po lokacijah oz. nahajališčih.
- Rešitev aplikacije naj bo čim bolj splošna tako, da bi jo bilo mogoče uporabiti tudi na drugih področjih popisa zalog, na primer v skladišču, itd.
- Aplikacija naj bo kar se da preprosta in omogoča intuitivno uporabo, kolikor je to le mogoče na majhnem zaslonu ročnega terminala.
- Artikli so označeni z črtno kodo tipa »Code 128-b« katera vsebuje številko artikla, ki ima alfanumerične znake (Slika 4.1).



Slika 4.1 Primer zahtevane črtne kode

- Inventura se izvaja po lokacijah, ki so prostorsko velike in niso označene z črtno kodo.
- Mora vsebovati možnost izbire lokacije iz vnesenega šifrantu lokacij.
- Inventuro popisuje več popisovalcev hkrati, zato naj bo v seznamu popisa razvidno kdo je kaj in kdaj popisal.
- Omogočati mora prikaz seznama popisanih artiklov za trenutnega prijavljenega popisovalca tako, da si ta seznam lahko ogleda revizor oz. nadzornik inventurne popisne komisije.
- Ker se artikel z isto šifro lahko ponovno najde tudi na drugi lokaciji, naj aplikacija na to opozori popisovalca in mu ponudi spremembo želene količine (povečanje ali zmanjšanje količine artikla).
- Aplikacija mora delovati na ročnem računalniku z operacijskim sistemom Windows Mobile 5.0 in 6.5 (Symbol oz. Motorola MC9090K in MC9590K).
- Lokacije popisa so dobro pokrite z WLAN omrežnim signalom.

### 4.3 Arhitektura

Za delovanje aplikacije potrebujemo dostop do WLAN omrežja. Ta je lahko brez internetne povezav ali pa z njo. Odvisno od tega kje se bo nahajala vaša podatkovna baza, ali na lokalnem strežniku ali na internetu. Aplikacija se preko HTTP zahtev poveže z strežnikom, ki omogoča *Web Service*, ta pa naprej povezuje na MySQL podatkovno bazo (Slika 4.2).



Slika 4.2 Arhitektura aplikacije

## 4.4 Struktura podatkovne baze

Struktura podatkovne baze je zgrajena iz treh tabel, ki so nujne za delovanje aplikacije. V naslednjih podpoglavjih so te tabele opisane, na slikah pa so prikazane njihove strukture.

### 4.4.1 Tabela »popisnik«

V tabeli z imenom »popisnik« imamo seznam uporabnikov aplikacije, v našem primeru so to člani inventurne komisije. Struktura tabele je razvidna iz spodnje slike (Slika 4.3). V tabeli pa so uporabljena naslednja polja:

- id (identiteta)
- username (uporabniško ime za prijavo)
- password (geslo za prijavo)
- ime (ime popisovalca ali člana inventurne komisije)
- priimek (priimek popisovalca ali člana inventurne komisije)
- admin (ta pri prijavi ločuje skrbnika aplikacije od ostalih uporabnikov)

#	Stolpec	Vrsta	Pravilo za razvrščanje znakov	Atributi	Null	Privzeto	Dodatno
<input type="checkbox"/> 1	<b>id</b>	int(11)			Ne	Brez	AUTO_INCREMENT
<input type="checkbox"/> 2	<b>username</b>	varchar(40)	latin1_swedish_ci		Ne	Brez	
<input type="checkbox"/> 3	<b>password</b>	varchar(60)	latin1_swedish_ci		Ne	Brez	
<input type="checkbox"/> 4	<b>ime</b>	varchar(30)	latin1_swedish_ci		Ne	Brez	
<input type="checkbox"/> 5	<b>priimek</b>	varchar(30)	latin1_swedish_ci		Ne	Brez	
<input type="checkbox"/> 6	<b>admin</b>	tinyint(1)			Ne	Brez	

Slika 4.3 Struktura tabele popisnik



### 4.4.2 Tabela »lokacije«

V tabeli z imenom »lokacije« imamo seznam nahajališč popisnega blaga. Struktura tabele je razvidna iz spodnje slike (Slika 4.4). V tabeli pa so uporabljena naslednja polja:

- id (identiteta)
- lok\_stev (številka lokacije)
- lak\_naziv (naziv lokacije)
- aktivna (ta ločuje aktivne in neaktivne lokacije, popis je možen samo na aktivnih)

<div> <div>Prebrskaj</div> <div>Struktura</div> <div>SQL</div> <div>Iskanje</div> <div>Vstavi</div> <div>Izvozi</div> <div>Uvozi</div> </div>							
#	Stolpec	Vrsta	Pravilo za razvrščanje znakov	Atributi	Null	Privzeto	Dodatno
<input type="checkbox"/> 1	<u>id</u>	int(11)			Ne	Brez	AUTO_INCREMENT
<input type="checkbox"/> 2	lok_stev	varchar(40)	latin1_swedish_ci		Ne	Brez	
<input type="checkbox"/> 3	lok_naziv	varchar(40)	latin1_swedish_ci		Ne	Brez	
<input type="checkbox"/> 4	aktivna	tinyint(1)			Ne	Brez	

Slika 4.4 Struktura tabele lokacije

### 4.4.3 Tabela »popis«

V tabeli z imenom »popis« imamo seznam popisanelega blaga (artiklov, materiala, sredstev, itd.). Struktura tabele je razvidna iz spodnje slike (Slika 4.5). V tabeli pa so uporabljena naslednja polja:

- id (identiteta)
- artikel (oznaka popisanelega blaga)
- kolicina (količina popisanelega blaga)
- lokacija (lokacija na kateri je bilo blago popisano)
- username (uporabniško ime ali uporabnik, ki je blago popisal)
- datum (datum in čas popisanelega blaga)

<div> <div>Prebrskaj</div> <div>Struktura</div> <div>SQL</div> <div>Iskanje</div> <div>Vstavi</div> <div>Izvozi</div> <div>Uvozi</div> </div>								
#	Stolpec	Vrsta	Pravilo za razvrščanje znakov	Atributi	Null	Privzeto	Dodatno	
<input type="checkbox"/>	1 <b>id</b>	int(11)			Ne	Brez	AUTO_INCREMENT	
<input type="checkbox"/>	2 <b>artikel</b>	varchar(60)	latin1_swedish_ci		Ne	Brez		
<input type="checkbox"/>	3 <b>kolicina</b>	float			Ne	Brez		
<input type="checkbox"/>	4 <b>lokacija</b>	varchar(60)	latin1_swedish_ci		Ne	Brez		
<input type="checkbox"/>	5 <b>username</b>	varchar(60)	latin1_swedish_ci		Ne	Brez		
<input type="checkbox"/>	6 <b>datum</b>	datetime			Ne	Brez		

Slika 4.5 Struktura tabele popis

## 4.5 Kriptiranje gesla

Zaradi varnosti prijavljanja v aplikacijo, sploh v primeru, če le ta deluje preko internetnega omrežja, smo za varovanje gesla proti prisluškovanju oziroma prestrezanju podatkov uporabili vgrajeni MD5 hash algoritem. Ta je že vgrajen v C izvorni kodi.

Kriptiranje gesla se v aplikaciji zgodi vedno kadar vnašamo geslo, in sicer:

- ob dodajanju novega uporabnika v seznam le teh
- ob spreminjanju gesla obstoječega uporabnika
- ob vnosu gesla ob prijavi

## 4.6 Spletni vmesnik

Spletni vmesnik (ang. *Web service*) smo razvili z programskim orodjem *Eclipse for Java EE developers*. Uporabili smo *JDK* različico 7.0. Odprli smo nov *Maven* projekt pri katerem smo kot tip izbrali *webapp* (slo. spletna aplikacija). *Maven* je v bistvu avtomatizirano orodje za gradnjo *Java* projektov. Uporablja se tudi za lažje testiranje in prenosljivost kode [20]. Spletno aplikacijo smo nato naložili na strežnik *Tomcat*, na spletno stran *Heroku*.

## 4.7 Predpogoji za delovanje aplikacije

V tem podpoglavju smo našli predpogoje za uporabo aplikacije, ki pa so:

- Naprava mora imeti nameščen operacijski sistem Windows Mobile 5.0 ali višji.
- Zelo priporočljivo je, da naj ima naprava vgrajen ali vsaj priključen bralnik črtne kode. Aplikacijo lahko uporabljamo tudi brez bralnika črtne kode, vendar to ne priporočamo saj bo popis artiklov veliko zamudnejši.
- Za uporabo bralnika črtne kode mora biti na napravi naložen ustrezen program, pripomoček, ki le tega aktivira (primer: DataWedge).
- Lokacije ali nahajališča pri popisu morajo biti z pokrita signalom WLAN omrežja.
- V omrežju mora biti dosegljiv računalnik ali strežnik z SQL podatkovno bazo z potrebnimi funkcionalnostmi za povezavo z le to.



## Poglavje 5 Uporaba mobilne aplikacije

V tem poglavju je predstavljena konkretna uporaba mobilne aplikacije tako s slikovnimi prikazi, kot tudi obrazložitvijo vseh vgrajenih funkcionalnosti. V grobem so funkcionalnosti aplikacije naslednje:

- preizkus povezave z podatkovno bazo
- prijava v aplikacijo
- šifrant lokacij
- šifrant uporabnikov
- popis (glavni del)
- pregled seznama popisa

Potrebna znanja za uporabo aplikacije so osnovno rokovanje z mobilnimi napravami in upravljanje zaslona na dotik. Aplikacijo namestimo na mobilno napravo, dodamo bližnjico na želeno mesto.

Na vseh vnosnih zaslonih v aplikaciji so vgrajene ustrezne kontrole vnosa. Aplikacija nam tako ne dovoljuje nepopolnega ali neustreznega vnosa v bazo podatkov in nam ob tem s kratkim obvestilom na zaslon sporoči napako. V primeru puščenega praznega polja nas tudi po zaprtem sporočilu usmeri nazaj na polje, ki smo ga pustili praznega. Pri vnašanju podatka v polje »količina« so dodane kontrole, ki ob tipkanju preverjajo ustreznost znakov. Dovoljene so samo numerične vrednosti od 0 do 9, ena decimalna pika ter predznak minus.

### 5.1 Testiranje aplikacije

Za hitro testiranje aplikacije smo uporabili vgrajeni simulator ročnega računalnika, in sicer *Windows Mobile 5.0 Pocket PC Emulator*, ki je vgrajen v programskem orodju *Visual Studio*. Predstavitvene slike aplikacije smo v tem poglavju uporabili kar iz omenjenega simulatorja. Za končno, bolj realno testiranje, smo uporabili ročni računalnik za katerega je bila tudi ta aplikacija prvotno razvita. To je bil ročni računalnik znamke Symbol, model

MC9090K, ki ima poleg ostalih funkcionalnosti vgrajen tudi bralnik črtne kode. Za bolj realen in hitrejši popis smo si izpisali nekaj pripravljenih črtnih kod tipa Code128, ki je bil določen pri zahtevah aplikacije. Te smo generirali z brezplačnim generatorjem črtne kode na spleti strani [2]. Vsaka izpisana črna koda je vsebovala svojo številko artikla s katerimi smo nato simulirali izvedbo inventure oz. popis artiklov.

## 5.2 Nastavitev povezave do baze podatkov

Preden se prvič prijavimo v aplikacijo naj oseba, ki bo zadolžena kot skrbnik aplikacije najprej preizkusi delovanje povezave do baze podatkov. S klikom na spodnji gumb »Nastavitve« nam aplikacija odpre zaslon nastavitve (Slika 5.1), v katerem nam nato ponuja možnost izbire in shranitve privzete poti do vmesnika *Web Service*, ki nam omogoča komuniciranje s podatkovno bazo.

Trenutno sta v aplikaciji nastavljeni dve različni povezavi med katerima lahko izbiramo in sta zapisani kar v kodi programa. Obe nastavitvi smo uporabljali pri testiranju aplikacije. Prva povezava kaže na *Web Service* računalnika v lokalnem omrežju. Druga povezava pa kaže na *Web Service*, ki smo ga naložili na eno izmed spletni strani. Z gumbom »Preizkus povezave WS« pa lahko preizkusimo delovanje trenutno izbrane povezave. Če povezava deluje se nam v tekstovno polje izpiše prebrani podatek iz baze, v nasprotnem primeru pa se nam izpiše obvestilo z napako, da povezava ni vzpostavljena.



Slika 5.1 Prijavni zaslon (levo) in zaslon nastavitvev (desno)

### 5.3 Prijava v aplikacijo

Prijava v aplikacijo ločuje med prijavljanje uporabnikov, ki samo izvajajo popis artiklov ter uporabnika, ki je lahko tudi skrbnik aplikacije. Skrbnik aplikacije ima vgrajene dodatne možnosti, kot so upravljanje šifranta popisnih lokacij in šifranta uporabnikov aplikacije. O tem več v nadaljevanju. Podatek kdo od uporabnikov je ali pa ni skrbnik aplikacije je zapisan v tabeli »popisnik«, in sicer v polju z imenom »admin« kar je kratica za administrator (slo. skrbnik).

V aplikacijo se prijavimo tako, da v ustrezna polja na prijavnem zaslonu vtipkamo svoje uporabniško ime in geslo (Slika 5.2) ali pa recimo ta dva podatka v polja preberemo kar z vgrajenim bralnikom črtna kode. Seveda v primeru, da imamo prej na nekakšni registracijski kartici natisnjeni črtni kodi s tema dvema podatkoma. Polje za geslo je zaradi varnosti nastavljeno tako, da pri tipkanju ne vidimo gesla in se za vsak vneseni znak namesto vtipkanega znaka prikaže znak \*. Geslo je v ozadju zaradi varnosti tudi kriptirano.



Slika 5.2 Prijava v aplikacijo

## 5.4 Glavni zaslon »Popis«

Po uspešni prijavi se nam odpre glavni zaslon »Popis«, v katerem se izvaja glavna dejavnost aplikacije, to je inventurni popis. V primeru, da smo se v aplikacijo prijavili kot uporabnik, ki ni skrbnik aplikacije, se nam v meniju aktivira samo možnost izhod iz aplikacije. V kolikor pa smo se v aplikacijo prijavili kot skrbnik aplikacije pa se nam aktivirajo oz. poudarijo še dodatne dve možnosti: »Uporabniki« in »Lokacije« (Slika 5.3).



Slika 5.3 Prikaz zaslona oz. menija uporabnika (levo) in skrbnika (desno)



### 5.4.1 Popis artikla na izbrano lokacijo

Preden začnemo z odčitavanjem črtnih kod, moramo najprej izbrati lokacijo popisa, na katero želimo popisati artikle, ki se na tej lokaciji nahajajo. Seznam lokacije se naloži iz šifranta lokacij oz. iz podatkovne tabele lokacije. Na sliki (Slika 5.4) je prikazan izbor lokacije in primer izpolnjenih polj za popis artikla. Z pritiskom na gumb »Shrani« ali s potrditvijo na tipko »Enter« (na tipkovnici) se podatki (artikel, količina, lokacija, popisnik in datum s časom popisa) shranijo bazo (v tabelo popis).

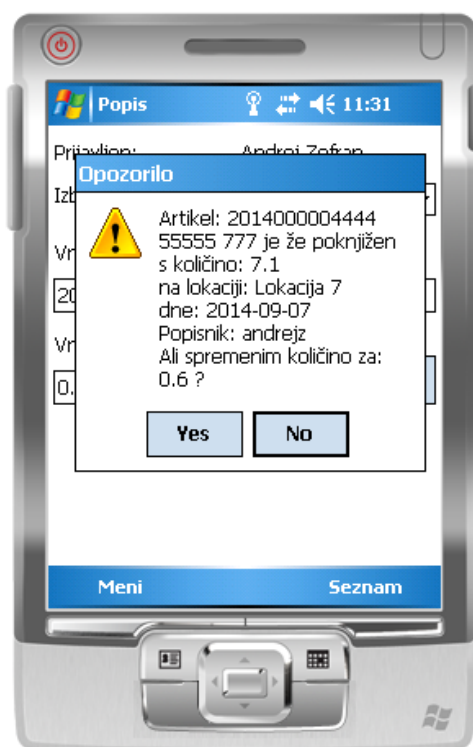


Slika 5.4 Izbor lokacije (levo) in primer popisa (desno)

### 5.4.2 Ponoven popis istega artikla

Ker je bila za aplikacijo podana zahteva, da mora omogočati dodajanje količine istega, že popisanega artikla, smo to funkcionalnost v aplikaciji rešili tako, da pri potrditvi popisa artikla preverimo ali je ta že bil popisan ali ne. V primeru, da je že bil popisan se na zaslonu prikaže sporočilo (Slika 5.5), ki uporabnika obvesti, da je artikel že bil popisan. Na obvestilu se prikažejo vsi podatki o popisu. Kdo je že popisal ta artikel, kje in kdaj je bil prvotno popisan ter s kakšno količino. Na obvestilu se prav tako ponudi vprašanje, ali vneseno količino spremenimo (v primeru pozitivnega števila se količina poveča, v primeru negativne količine pa ta zmanjša).

Funkcionalnost ponovnega popisa istega, že popisanega artikla, lahko uporabimo tudi za popraviljanje napak. Na primer, da smo za artikel po pomoti vpisali preveliko količino. Tako lahko tega ponovno popišemo z negativno količino. To storimo tako, da v polju pred količino uporabimo negativen predznak (minus). Po potrditvi se bo tako količina v bazi pomanjšala za želeni znesek. V primeru, da je prišlo ob popisu do napačno izbrane lokacije lahko to tudi popravimo. To storimo tako, da izberemo novo lokacijo, ponovno odčitamo artikel in vpišemo količino 0. Tako se bo v bazo ažurirala samo sprememba lokacije.



Slika 5.5 Obvestilo o ponovnem popisu že popisanega artikla

## 5.5 Pregled seznama popisa

Na glavnem zaslonu imamo še možnost prikaza seznama že popisanih artiklov, ki je dosegljiva preko gumba »Seznam«. S klikom nanj se nam odpre nov zaslon, na katerem se izpiše seznam vseh artiklov, ki jih je prijavljeni uporabnik do tega trenutka popisal (Slika 5.6). Ker so običajno pri popisu prisotni tudi revizorji je glavni namen tega seznama ta, da služi za kontrolo in nadzor izvajanja popisa oz. dela inventurne komisije. Preko tega seznama lahko popisovalec pokaže revizorju seznam popisanih artiklov, ter kje in kdaj jih je popisal.

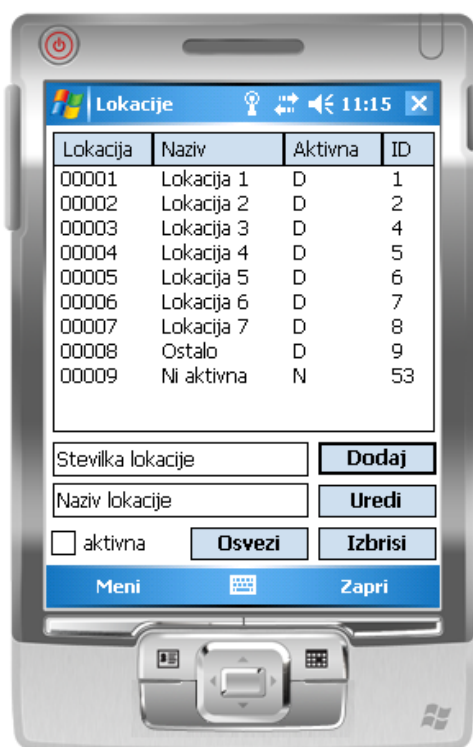


Slika 5.6 Seznam popisa

## 5.6 Šifrant lokacij

Lokacije imajo v aplikaciji funkcijo nahajališč na katerih se nahajajo artikli v trenutku, ko izvajamo inventuro. Na zaslonu lokacije je tako prikazan šifrant oz. seznam lokacij, katerega lahko poljubno urejamo. Zaslona nam ponuja funkcionalnosti kot so: dodajanje nove, urejanje in brisanje obstoječe lokacije (Slika 5.7).

Do zaslona lokacije lahko preko menija v glavnem zaslonu »Popis«, dostopa le uporabnik aplikacije, ki je opredeljen tudi kot skrbnik oz. administrator aplikacije.

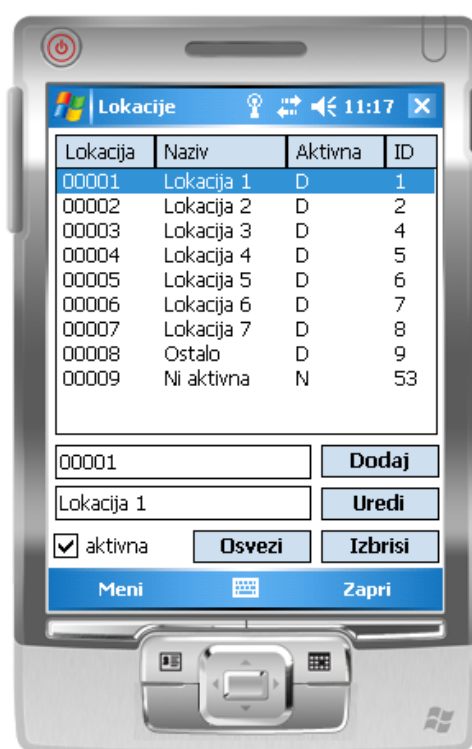


Slika 5.7 Šifrant lokacij

### 5.6.1 Dodajanje in urejanje lokacije

V primeru, da želimo v seznam oz. šifrant dodati novo lokacijo, najprej izpolnimo oba polja (številka lokacij, naziv lokacije) in označimo kvadratega aktivna ter kliknemo na gumb dodaj. Če lokacijo ne označimo kot aktivno pomeni, da se ta ne prikaže iz listajočega seznama lokacij na glavnem popisnem zaslonu in na njo tako ni mogoče izvajati popisa artiklov. V kolikor lokacijo označimo kot aktivno, se ta prikaže na omenjenem listajočem seznamu in na njo lahko popisujemo artikle.

V primeru urejanja lokacije pa to najprej izberemo iz prikazanega seznama. S tem se polja napolnijo s podatki o izbrani lokaciji in te lahko preprosto uredimo ter potrdimo spremembe s klikom na gumb »Uredi« (Slika 5.8).



Slika 5.8 Izbrana lokacija

## 5.7 Šifrant uporabnikov

Glavni namen šifranta uporabnikov je ta, da se le ti pri vsakem popisu artikla zavedejo v bazo podatkov oz. na seznam popisa. Tako lahko za vsakega popisovalca vidimo katere artikle je popisal, s kakšno količino, na kateri lokaciji so bili popisani in kdaj so bili popisani (datum in čas). S tem lahko tako kontroliramo njegovo delo, kot tudi preverimo njegovo učinkovitost pri izvedbi inventure.

Na zaslonu »Uporabniki« je prikazan seznam uporabnikov katerega lahko poljubno urejamo. Zaslون nam ponuja funkcionalnosti kot so: dodajanje novega, urejanje in brisanje obstoječega uporabnika (Slika 5.9).

Prav tako kot do zaslona »Lokacij«, tudi do šifranta uporabnikov, lahko preko menija v glavnem zaslonu »Popis« dostopa le uporabnik aplikacije, ki je opredeljen tudi kot skrbnik aplikacije.

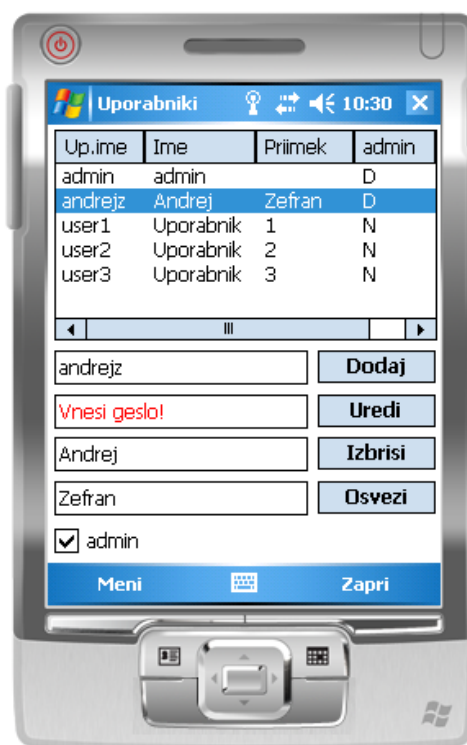


Slika 5.9 Šifrant uporabnikov

### 5.7.1 Dodajanje in urejanje uporabnika

V primeru, da želimo v seznam dodati novega uporabnika, najprej izpolnimo vsa polja (uporabniško ime, geslo, ime, priimek), ter kliknemo na gumb »Dodaj«. Če želimo uporabnika zavesti kot skrbnika aplikacije, to označimo s klikom v polje »admin«.

V primeru urejanja uporabnika, prav tako kot pri uporabi šifranta lokacij, tega najprej izberemo iz prikazanega seznama. S tem se polja napolnijo s podatki izbranega uporabnika. Pri tem se geslo zaradi varnosti oz. kriptiranja podatkov ne prikaže na zaslonu, lahko pa ga popravimo enako kot ostale podatke in s klikom na gumb »Uredi« preprosto potrdimo vse spremembe. (Slika 5.10).



Slika 5.10 Izbrani uporabnik





## **Poglavje 6      Sklepne ugotovitve**

V diplomskem delu smo razvili in opisali mobilno aplikacijo za izvedbo inventure z ročnim računalnikom z vgrajenim bralnikom črtne kode. Aplikacija je zasnovana na operacijskemu sistemu Windows Mobile. Uporabili smo razvojni paket Microsoft Visual Studio 2008. Za identifikacijo blaga smo uporabili črtno kodo, ki je zelo razširjena in preprosta za uporabo.

Cilj dela je bil razviti aplikacijo z namenom odprave zamudnega ročnega popisa inventure in tako bistvene pohitritve tega procesa, ki se še dandanes v marsikaterem podjetju izvaja ročno. Elektronski način je za razliko od ročnega enostavnejši, zanesljivejši in predvsem dosti hitrejši ter omogoča tudi boljšo kontrolo in preverjanje učinkovitosti popisovalcev.

Z aplikacijo je tako možno izvesti inventuro popisa zalog nedovršene proizvodnje, skladiščnih zalog surovin in materiala ter raznega blaga tudi na ostalih področjih, kjer se izvaja inventura. Da bo popis čim bolj učinkovit, natančen in hiter je najprimerneje, da je blago označeno s črtno kodo, katero podpira tudi sam ročni terminal. Aplikacija omogoča shranjevanje popisnih podatkov večjega števila popisovalcev hkrati, tako na istih, kot tudi na različnih lokacijah.

Kljub temu, da ima mobilna aplikacija kar nekaj vgrajenih funkcionalnosti ima tudi še vsaj dve slabosti, katere bi lahko še izboljšali. Prva slabost je da potrebuje brezžično omrežje WLAN. Druga pa, da lokacij trenutno ni možno prebrati z bralnikom črtne kode, v kolikor bi te bile v podjetju ustrezno označene.

Smer nadaljnjega razvoja aplikacije bi lahko bila naslednja:

- Omogočiti izvedbo inventure tudi brez WLAN omrežja, in sicer na način, da bi se podatki shranjevali v lokalno bazo ali datoteko na mobilnem računalniku ter bi to kasneje preko USB, Bluetooth, WLAN ali drugačne podobne povezave sinhronizirali v skupno bazo podatkov.
- Za izbor lokacije popisa bi v aplikacijo poleg izbire iz listajočega seznama dodali še možnost branje lokacije z bralnikom črtne kode.
- Aplikacijo bi še vizualno nadgradili, v očesu privlačnejšo z raznimi grafičnimi orodji.

- Aplikaciji bi lahko dodali še različna signalna opozarjanja. Na primer različen zvok v primeru neuspešnega ali uspešnega popisa oz. drugačen zvok v primeru raznoraznih napak, ki jih javlja aplikacija v primeru napačne uporabe le te.
- Izvoz podatkov iz podatkovnih baz glede na potrebe.

Kljub naštetim pomanjkljivostim je aplikacija enostavna za uporabo in vključuje vse zahteve, ki smo si jih pred razvojem zastavili.

## Literatura

- [1] Bart A. DePetrillo, Razumeti Microsoft .NET, str. 50-53, 2002.
- [2] Brezplačni spletni generator črtne kode [Online]. Dosegljivo:  
[http://www.barcoding.com/upc/#.VA8iqPl\\_tmN](http://www.barcoding.com/upc/#.VA8iqPl_tmN)
- [3] DataWedge user guide, *MotorolaSolutions* [Online]. Dosegljivo:  
[https://launchpad.motorolasolutions.com/documents/dw\\_user\\_guide.html](https://launchpad.motorolasolutions.com/documents/dw_user_guide.html)
- [4] Črtna koda: tipi (simbiologije), *Leoss* [Online]. Dosegljivo:  
<http://www.leoss.si/index.php?vie=ctl&gr1=strSvt&gr2=&id=2008061009152643>
- [5] Črtna koda, *Wikipedia* [Online]. Dosegljivo:  
[http://sl.wikipedia.org/wiki/%C4%8Crtna\\_koda](http://sl.wikipedia.org/wiki/%C4%8Crtna_koda)
- [6] Eclipse, [Online]. Dosegljivo: <http://www.eclipse.org/org/>
- [7] Inventura ali popis, *eracunovodstvo.org* [Online]. Dosegljivo:  
<http://www.eracunovodstvo.org/blog/tag/popis/>
- [8] James F. Kurose, Keith W. Ross, Computer Networking (fifth edition), str. 713-733, 2010.
- [9] Java EE, [Online]. Dosegljivo:  
[http://en.wikipedia.org/wiki/Java\\_Platform,\\_Enterprise\\_Edition](http://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition)
- [10] Kaj je črtna koda?, *Leoss* [Online]. Dosegljivo:  
<http://www.leoss.si/index.php?vie=cnt&gr1=fr&gr2=&gr3=&id=2006082214500003&lng=slo>
- [11] Kriptografija, *Wikipedia* [Online]. Dosegljivo:  
<http://en.wikipedia.org/wiki/Cryptography>
- [12] Microsoft Visual Studio, *Wikipedia* [Online]. Dosegljivo:  
[http://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio)

- [13] Mobilni operacijski sistem, *Wikipedia* [Online]. Dosegljivo: [http://en.wikipedia.org/wiki/Mobile\\_operating\\_system](http://en.wikipedia.org/wiki/Mobile_operating_system)
- [14] Motorola Symbol MC9500, *Info-kod* [Online]. Dosegljivo: <http://www.info-kod.si/si/izdelki-in-resitve/ročni-terminali/windows/motorola-symbol-mc9500>
- [15] MySQL, [Online]. Dosegljivo: <http://dev.mysql.com/doc/refman/4.1/en/what-is-mysql.html>
- [16] Ročni računalniki, *Leoss* [Online]. Dosegljivo: <http://www.leoss.si/index.php?lng=slo&vie=prod&id=2005090209453003&var1=32>
- [17] Ročni terminal Motorola MC9000 *Leoss* [Online]. Dosegljivo: <http://www.leoss.si/index.php?lng=slo&vie=prod&id=2005090209393127&var1=32>
- [18] Simbologija CODE 128, *Leoss* [Online]. Dosegljivo: <http://www.leoss.si/index.php?vie=cnt&gr1=nvc&gr2=&gr3=&id=2008061008465804&lng=slo>
- [19] Visual C# . NET, [Online]. Dosegljivo: <http://uranc.tsckr.si/VISUAL%20C%23/VISUAL%20C%23.pdf>
- [20] Welcome to Apache Maven, [Online]. Dosegljivo: <http://maven.apache.org/>
- [21] Windows Mobile, *Wikipedia* [Online]. Dosegljivo: [http://en.wikipedia.org/wiki/Windows\\_Mobile](http://en.wikipedia.org/wiki/Windows_Mobile)
- [22] .NET Framework, *Wikipedia* [Online]. Dosegljivo: [http://en.wikipedia.org/wiki/.NET\\_Framework](http://en.wikipedia.org/wiki/.NET_Framework)